UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

TRABAJO FIN DE MÁSTER

"Aplicación de Algoritmos de Machine Learning para predecir la probabilidad de abandono de estudiantes en MOOCs basándose en el modelo Context-aware Feature Interaction Network"

Máster Universitario en Ingeniería Informática

Autor: DIOS LUNA, Jim Bryan

Tutora: CARRO SALAS, Rosa María
Departamento de Ingeniería Informática

Febrero, 2021

Índice General

Índice de Figuras

Índice de Tablas

Índice de Bloque de Código

Resumen

Desde su aparición hace algunos años, los MOOC (Massive Online Open Courses) han representado una gran oportunidad de formación académica a nivel superior, puesta al alcance de millones de estudiantes alrededor del mundo, quienes en la mayoría de los casos no hubiesen podido acceder a este tipo de conocimiento por las limitaciones que brinda una educación presencial, principalmente por el coste del estudio, estadía o transporte que esta implica.

Pese a ello, y sobre todo en un mundo en el que existe la necesidad de obtener nuevos conocimientos para hacer frente a un entorno profesional y laboral cada vez más competitivo, aún se espera que los MOOC tengan un mayor impacto en la sociedad del conocimiento. Muchos estudios han evidenciado la baja tasa de finalización por parte de los estudiantes frente a un abundante número de abandono de los mismos, lo que hace propicio abordar este problema desde un enfoque de Learning Analytics, el cual nos brinda técnicas y herramientas que permiten analizar el comportamiento registrado en los MOOC por parte de los estudiantes para fortalecer las estrategias en las que se ha basado un curso, y, recomendaciones generadas hacia los actores, profesores y estudiantes respecto del progreso que van realizando.

El valor que se genera a través de los registros (logs) de las acciones de un estudiante de un MOOC es incalculable al momento de proponer alguna alternativa de solución, pues es en ellos donde se almacena cada interacción generada por los estudiantes, la misma que nos permitirá identificar el comportamiento durante el proceso de aprendizaje e intentar predecir (y, por tanto, prevenir) posibles abandonos en un MOOC.

Durante este Trabajo de Fin de Máster, se han realizado procesos de análisis exploratorio y preprocesamiento de los datos para extraer la información más relevante de las interacciones registradas por los estudiantes en los MOOCs ofertados por la plataforma XueTangX, específicamente en las categorías de computación, economía e ingeniería, y convertirla en el insumo para la aplicación de algoritmos de aprendizaje supervisado y redes neuronales, para construir y comparar modelos de clasificación que permitan determinar si un estudiante abandonará o no un MOOC.

Abstract

Since their appearance a few years ago, MOOC (Massive Online Open Courses) have represented a great opportunity for high-level academic training, made available for a great number of students around the world who, in the majority of cases, would not have been able to access this kind of knowledge due to limitations that access to face-to-face education, and in many cases paid, implies.

Despite this, and primarily in a world in which the idea of obtaining new knowledge to face an increasingly competitive professional and work environment, MOOC are still expected to have a greater impact on knowledge society as many studies have shown a low completion rate by enrolled students in comparison with a large number of dropouts. What makes it appropriate to address this problem from a Learning Analytics approach, as it provides us with techniques and tools that allow students to analyze behavior recorded in MOOC to strengthen the strategies on which a course has been based, recommendations to the actors, professors, and students regarding the progress they have made.

In this way, the value of logs of MOOC students is enormous when proposing a solution alternative, since it is in them where every interaction generated by the students is stored, the same as it will allow us to identify behavior during the learning process and try to predict (and therefore prevent) possible dropouts in a MOOC.

For this Master's Dissertation, exploratory data analysis and preprocessing have been carried out to extract the most relevant information from the interactions recorded by students in the MOOCs offered by the XueTangX platform, specifically in the categories of computing, economics, and engineering, and turn it into the input for the application of supervised learning algorithms and neural networks, to build and compare classification models that determine whether or not a student will drop out of a MOOC.

1. Introducción

1.1. Motivación

Durante las últimas décadas, la educación, y en especial la educación superior, ha demostrado ser de vital importancia para los avances en la sociedad, no solo desde el plano científico, en el que numerosas investigaciones han aportado grandes logros en campos como la medicina y la industria, sino también y sobre todo en tecnología, hecho que ha generado una gran revolución y ha sumergido a casi la totalidad de los seres humanos en el uso de la misma, pues, entre otros usos, nos permite conectarnos en cuestión de segundos con personas que se encuentran a miles de kilómetros de distancia pero cercanas digitalmente.

Es así que la educación, durante la época de la ilustración, sólo era accesible para gente perteneciente a las clases sociales más altas; sin embargo, y con el transcurrir del tiempo y tras los diferentes esfuerzos aportados por distintas sociedades, se ha ido universalizando para convertirse finalmente en una herramienta más del desarrollo del mundo globalizado en el que hoy vivimos.

Durante las épocas de los años 70, 80, y parte de los 90 las revoluciones industriales demandaban el dominio de conocimientos superiores ofrecidos por las escuelas u otros centros de formación, generando gran demanda de profesionales, quienes por diversas razones aún no habían alcanzado el conocimiento técnico requerido para aportar ese conocimiento especializado a las grandes empresas, fábricas y otros espacios donde fueran demandados. Dicha necesidad llevó a idear una forma accesible para quienes no tuvieran la oportunidad de asistir presencialmente a las clases impartidas en institutos o universidades, y es donde aparece la opción de estudiar bajo una modalidad no presencial o a distancia lo que significó un hito para la diversificación y acceso a estudios superiores para muchas personas; no obstante, como en todo proceso de aprendizaje, siempre han existido factores de riesgo, que conllevan que los estudiantes no culminen los estudios que inician. Esos factores se encuentran entre los que más interesaban a los pedagogos y grupos de académicos, quienes buscaban garantizar la culminación de estos beneficiosos programas. Lastimosamente, realizar un seguimiento e identificación de dichos factores resultaba sumamente costoso en todos los aspectos, desde el monitoreo de los estudiantes, pasando por la recolección y envío de información, así como los

medios para procesar y entender la situación de cada estudiante oportunamente y, eventualmente, apoyarlos en terminar los estudios iniciados.

Por tanto, como ya se ha señalado, el gran aliado de la educación ha sido y seguirá siendo la tecnología, y con ello la llegada los Massive Online Open Courses (MOOC), que posibilitan la reducción de brechas en la sociedad, pues han cambiado la forma de aprendizaje de muchos estudiantes, al pasar de tener un aula física limitada en espacio y tiempo, a tener miles de estudiantes ubicados en diferentes partes del mundo aprendiendo a su propio ritmo.

Después de algunos años, aún se espera que estos MOOCs terminen por llegar a su máximo nivel de aprovechamiento por los estudiantes, dado que en la actualidad se cuenta con muchas herramientas de proceso y análisis de datos que permiten identificar lo que en un pasado no se pudo hacer. En ese sentido, encontramos en la analítica del aprendizaje (Learning Analytics) una herramienta que, entre otras cosas, permite la detección y prevención de abandono de estudiantes mediante el uso de algoritmos de aprendizaje automático. Considerando las investigaciones existentes, se plantea la necesidad de complementarlas aportando una nueva forma de análisis que permita principalmente predecir un posible abandono del MOOC en el que se haya apuntado un estudiante, teniendo en cuenta datos demográficos, el comportamiento del estudiante así como la relación entre sus compañeros en la misma condición, los cuales podrían motivar a un abandono en bloque de los cursos MOOC, considerando el modelo Context-aware Feature Interaction Network. La realización de este trabajo se sitúa en este contexto y para desarrollarlo se han puesto en práctica las habilidades desarrolladas durante el estudio de las materias del Máster Universitario en Ingeniería Informática.

1.2. Objetivos

1.2.1 Objetivo principal

El objetivo general de este proyecto es explorar la posibilidad de construir un modelo basado en algoritmos de Aprendizaje Automático para predecir un posible abandono de un curso de un estudiante, analizando sus patrones de comportamiento en el desarrollo de materias, así como la relación de sus datos en los cursos en comparación con los de otros estudiantes.

1.2.2 Objetivos particulares

El objetivo general se desglosa en los siguientes objetivos concretos:

O.1 Clasificación de estudiantes por grupos según su comportamiento en los grupos de materias analizados.

O.2 Establecimiento de correlación entre grupos de estudiantes con probabilidad de abandono de materias.

O.3 Elaboración de un modelo con el que se trate de predecir la intención de abandono de un estudiante de algún curso basado en las interacciones registradas con respecto al mismo.

1.3. Contexto del Trabajo de Fin de Máster

Con el objeto de analizar la mayor cantidad de información posible, se trabaja con los datos recopilados por una plataforma educativa de Asia (XueTangX). A partir de esta información se trata de identificar el comportamiento de los estudiantes, medir su tasa de abandono de los cursos y tratar de identificar el motivo de dicho abandono, considerando, entre otras cosas, el tipo y la cantidad de actividades realizadas durante el desarrollo del curso, asociándolas también a datos demográficos, como edad, nivel de educación y género.

1.4. Estructura de la Memoria

De aquí en adelante, la memoria se estructura en los siguientes capítulos. El capítulo 2 recoge el estado del arte sobre las herramientas de Learning Analytics. El capítulo 3 describe el análisis de los datos. El capítulo 4 presenta los resultados obtenidos de la aplicación de algoritmos de aprendizaje supervisado para predecir un posible abandono. Finalmente, el capítulo 5 recoge las conclusiones de este trabajo y algunas ideas sobre posibles líneas futuras.

2. Estado del Arte

En el estado del Arte se habla de las nuevas tendencias propuestas en los últimos años para fortalecer las herramientas de Learning Analytics con la ayuda de nuevas tecnologías.

Hoy en día, el reto de Learning Analytics (LA) es entender cómo la teoría analítica implica pasar "*De los clicks a construir*" [1], dado que ya no es necesario diseñar y proponer un modelo teórico sobre cómo funcionan las cosas en el mundo, porque la era del petabyte-scale [2], es capaz de decirnos, a partir de los datos, qué es lo que está ocurriendo en tiempo real, el impacto de nuestras decisiones y los cambios que éstas generan.

J.Cong, propone analizar los datos del comportamiento del aprendizaje de los estudiantes de MOOC's, utilizando métodos de *feature extraction* para identificar las características del comportamiento del aprendizaje de los estudiantes semanalmente, para luego ser utilizadas en la construcción de un modelo basado en Support Vector Regression (SVR), el mismo que es utilizado como Student Dropout Prediction (SDP) *model* y cuyos parámetros son determinados por el algoritmo Improved Quantum Particle Swarm Optimization (IQPSO), de los que se obtienen mejores resultados frente a los alcanzados por los SDP [3].

X. Lu, S.Wang, J. Huang, W. Chen y Z. Yan, analizan un *dataset* de *logs* de estudiantes de MOOC's de la Universidad de Peking alojados en Coursera, de donde extraen las principales características de inicio y punto de abandono del MOOC, en esta investigación propone un modelo para predecir la probabilidades de abandono de algún curso, y por otro lado, un modelo para predecir si un estudiante obtendrá o no, una nota final en el curso. Brindado a los profesores y diseñadores de los cursos, información en la que basen mejoras en la calidad del curso en futuras versiones [4].

Por otro lado, LA brinda un gran potencial a educadores y aprendices, que son capaces por primera vez de ver su propio proceso y progreso de maneras que antes no era posible [5].

2.1. MOOCs

A diferencia del aprendizaje abierto o a distancia, los Massive Open Online Courses (MOOCs) han sido durante los últimos años un fenómeno relativamente nuevo que ha proporcionado muchas oportunidades de aprendizaje abierto y a distancia [6], poniendo al alcance de millones de estudiantes cursos de educación superior sin limitaciones de espacio o tiempo, en relación con los cursos dictados en un salón de clases de universidad, pues son accesibles desde Internet a cada momento, con acceso a diversos recursos como: textos, ilustraciones y contenidos de vídeo, también, pueden interactuar en foros del mismo curso y plantear consultas o establecer comunicaciones con otros estudiantes y profesores alrededor del mundo [6]. Según reporte de Class Central Report 2020[7], en ese año se estimó que Coursera [8], edX [9] y Future Learn [10], tenían aproximadamente unos 20 millones, 8 millones y 4 millones de nuevos usuarios registrados a nivel mundial y, un acumulado total de 65 millones, 32 millones y 13.5 millones de usuarios registrados, respectivamente.

Tomando en cuenta la cantidad de datos generada por los usuarios durante su participación en los cursos, resulta muy atractiva la idea de explorar y explotar dicha información aplicando técnicas de Learning Analytics.

Durante el desarrollo de este TFM se ha explorado la plataforma XueTangX [11], que dispone alrededor de unos 3000 cursos distribuidos en trece categorías con un aproximado de 15 millones de estudiantes [12].

2.2. Learning Analytics

Luego de haberse propuesto diferentes definiciones para la Analítica del aprendizaje (Learning Analytics) en sus inicios, la definición más utilizada proviene de la International Conference on Learning Analytics*, "La analítica del aprendizaje es la medición, colección, análisis y reporte de datos de estudiantes y sus contextos, con el fin de comprender y optimizar el aprendizaje y los entornos en los que ocurre" [13].* Directamente relacionados con LA se encuentran los campos del aprendizaje, tales como investigación educacional, ciencias del aprendizaje y evaluación, o tecnología educacional; analítica, tales como estadística, visualización, ciencia de datos/computación e inteligencia artificial); y diseño centrado en humanos, tales como usabilidad, diseño participativo y pensamiento de sistemas sociotécnicos.

Analítica del aprendizaje supone también el uso de los datos producidos por los alumnos, y el análisis de los mismos para descubrir la información y las relaciones sociales, con la finalidad de predecir y asesorar alguna acción relacionada con el aprendizaje [13]. Complementariamente a ello, no solo busca mejorar los procesos de formación y aprendizaje de los estudiantes mediante la identificación de patrones, sino que también ayudan a los profesionales de la educación a adaptar y a modelar su enseñanza, incrementándose así los beneficios del aprendizaje personalizado.

Considerando que el conjunto de datos que se desea analizar proviene de las interacciones realizadas por estudiantes en MOOCs, es importante resaltar los distintos métodos y metodologías asociadas al proceso de analítica del aprendizaje. A continuación, se describen 3 métodos computacionales:

- **Network-Analytic Methods**

  Basado en aproximaciones de análisis de redes (Network-Analytic), este enfoque se centra en las relaciones existentes entre los actores, los cuales, haciendo una analogía con un grafo, se representan mediante nodos, representándose sus relaciones con otros actores mediante arcos. Las conexiones entre los actores pueden ser de tipo: afiliación, amistad, profesional o por la información que comparten sobre algún interés en particular; es importante destacar que las técnicas de network analytic no necesariamente se basan en relaciones entre actores y relaciones sociales, sino que también se pueden representar relaciones entre los modos de aprendizaje de la plataforma (modo al ritmo del instructor y al propio ritmo) [14].

- **Process-Oriented Interaction Analysis**

  Este enfoque consiste en realizar un análisis computacional de las interacciones de los estudiantes que están registradas en un log durante el desarrollo de un curso [14]. Las distintas plataformas que dan soporte a los MOOCs permiten la extracción de esta información, y mediante un proceso de exploración y transformación de los datos se puede realizar este tipo de análisis de la interacción, directamente relacionado con el objetivo de este TFM.

- **Content Analysis Using Text-Mining Methods**

  Este enfoque está basado en el análisis del contenido textual generado por los estudiantes durante el desarrollo de los cursos, especialmente en los foros de discusión de donde se extrae información semántica que, ayudada de aproximaciones y modelos como "bag of words" [14], permite identificar oportunamente las relaciones de palabras de los diferentes temas abordados en un curso MOOC.

2.3. Metodología

En este trabajo se propone utilizar una metodología de analítica prescriptiva (*prescriptive analytics*), combinando la utilización del método computacional *Process-Oriented Interaction Analysis* con aprendizaje automático, para predecir una posible tasa de abandono de estudiantes en los MOOCs, en los cursos de economía, computación e ingeniería. A continuación, se describen las bases de esta metodología.

Machine Learning

Tom M. Mitchel define Machine Learning como "*El área del aprendizaje automático cuya preocupación es la construcción de programas informáticos que mejoren automáticamente con la experiencia… considerando que dicho programa informático aprende una experiencia E con respecto a alguna clase de tarea T y medida de rendimiento P, si su rendimiento en las tareas en T, medida por P, mejora con la experiencia E*" [15].

El desarrollo y crecimiento de la analítica del aprendizaje está fuertemente relacionado con el uso de algoritmos de aprendizaje automático, tales como árboles de decisión, redes neuronales, redes bayesianas y *clustering*, entre otros. Estos algoritmos son recurrentemente usados en distintas investigaciones y su característica principal es que se basan en la ejecución sobre grandes cantidades de datos, previamente procesados y etiquetados, con el fin de aprender y encontrar reglas de asociación, patrones ocultos y/o predecir comportamientos futuros.

En el ámbito educativo, los algoritmos pueden aprovechar toda la información disponible sobre interacciones de los estudiantes, registrada durante el proceso aprendizaje en un MOOC.

El aprendizaje automático, por tanto, consiste en la aplicación de algoritmos asociados a distintos métodos. En este trabajo se explorará el uso de diversos algoritmos utilizando conjuntos de datos (*datasets*) de estudiantes de MOOCs obtenidos de la plataforma de XuetangX [9] para comparar los modelos e identificar el de mayor precisión para predecir el riesgo de abandono de un estudiante de un MOOC, considerando su contexto.

Redes Neuronales Artificiales (ANN)

Son redes inspiradas en las interconexiones de nuestro cerebro biológico y, a diferencia de las Redes Neuronales Biológicas, tienen como diferencia el poder conectarse entre capas discretas considerando las direcciones de propagación de los datos, que les permiten realizar, entre otras tareas, la de clasificación, partiendo de imágenes, texto o sonido [16].

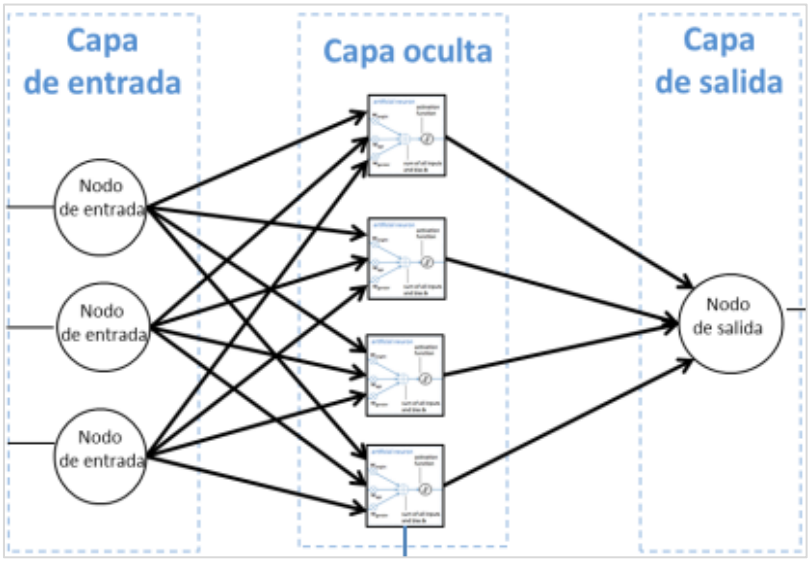

Figura 1: Neurona Artificial [16]

Para este trabajo se propone la utilización de una Red Neuronal Artificial utilizando el modelo secuencial (uso de una serie de capas de neuronas, "*una delante de otra*") que ofrece el módulo Keras, utilizando para las capas de entrada y oculta la función de activación de la Unidad Lineal Rectificada (ReLu), función que genera la entrada directa si es positivo o, de lo contrario, generará 0 en caso de ser negativo. Esta función de activación ha sido seleccionada por ser una de las más empleadas en la actualidad al comprobarse que tiene una mayor efectividad en los resultados, frente a otras funciones como Sigmoid y Tanh. Para la capa de salida se ha considerado la función de activación Sigmoid para garantizar que el resultado de la red sea de 0 y 1 [16] [17] [18].

3. Análisis de los datos

3.1. Conjunto de datos

Para la realización del Trabajo de Fin de Máster se ha utilizado un conjunto de archivos que corresponden a grupos de cursos ofrecidos por la plataforma de MOOCs XueTangX [9], los cuales se describen a continuación:

- **User_info**

  El archivo, en formato .csv, contiene información respecto al estudiante: id, género, educación y fecha de nacimiento.

- **Course_info**

  El archivo, en formato .csv, contiene información respecto al curso: id, nombre, inicio, fin, tipo (a propio ritmo, o a ritmo de profesor), categoría del curso.

- **Log_info**

  El archivo, en formato .csv, contiene el registro de las interacciones de un estudiante generadas en un curso: enrollamiento, nombre de usuario, id de curso, id de sesión, acción, tiempo.

- **Truth_info**

  El archivo, en formato .csv, contiene información respecto a la posibilidad de que un estudiante abandone un curso que está realizando (0=falso, 1=Positivo).

Los datos obtenidos del *dataset* han sido integrados considerando el campo enroll_id entre los *dataframe* de los archivos **Course_info, Log_info** y **Truth_info**. Así mismo, los datos de **User_info** han sido concatenados usando un método for.

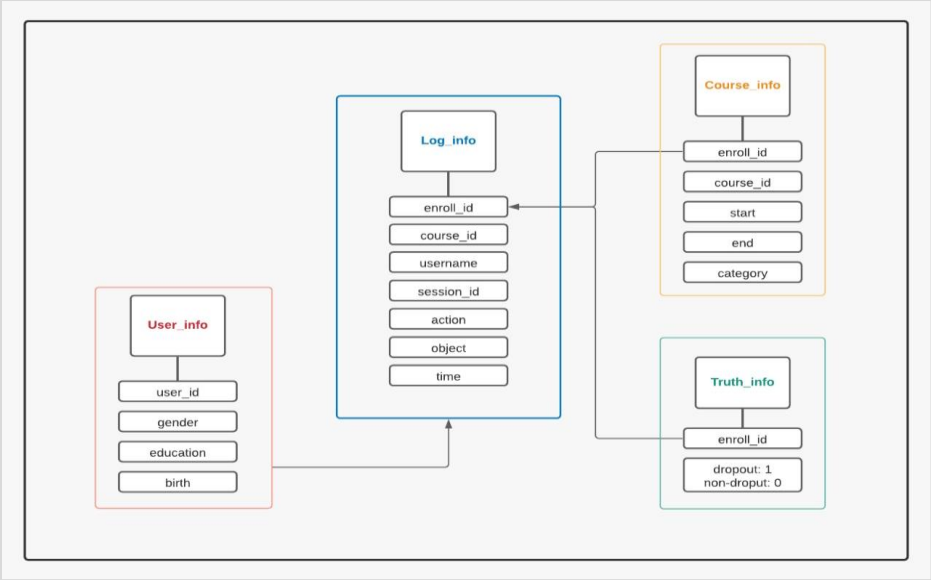

Figura 2: Diagrama de integración de los datos.

3.1.1 Estructura de los registros

Durante el desarrollo de este TFM, se ha considerado la interacción generada por los estudiantes en los diferentes cursos a los que tiene acceso dentro de la plataforma XuetangX, como un insumo de suma importancia para nuestro análisis de datos. Se verifica que la estructura del registro de un evento es la siguiente (ver figura 3): id del curso, id del usuario, sesión del usuario, actividad/evento y tiempo.

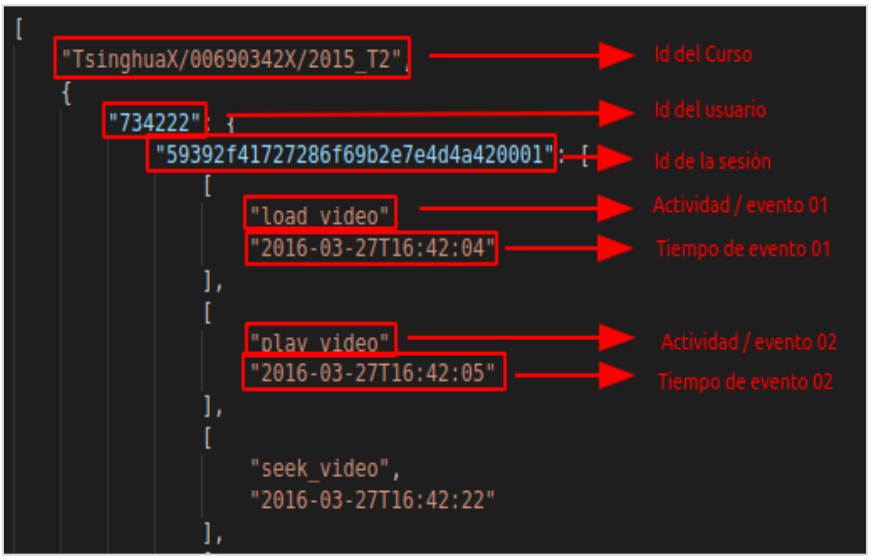

Figura 3: Estructura de registro de usuarios XueTangX

Existen 5 tipos de eventos, relacionados, respectivamente, con los vídeos, los problemas, foros, acciones de *click* y acciones de cierre, y un total de 21 acciones asociadas a estos eventos, tal y como se puede observar en la *Tabla 1*:

Evento	**Subcategoría**
video_action	seek_video, play_video, pause_video, stop_video, load_video
problem_action	problem_get, problem_check, problem_save, reset_problem, problem_check_correct, problem_check_incorrect
forum_action	create_thread, create_comment, delete_thread, delete_comment
click_actionforum_action	click_info, click_courseware, click_about, click_forum, click_progress
close_action	close_courseware

Tabla 1: Eventos y acciones asociadas

Para el caso de los eventos relacionados con los vídeos, se ha realizado un proceso de extracción de la cantidad de registros por cada tipo de evento. Es decir, se ha establecido una sumatoria del número de **seek / play / pause / stop / load** de cada vídeo. Este mismo procedimiento se ha llevado a cabo para los demás eventos y acciones.

Se cuenta también con un registro de los datos demográficos de los estudiantes registrados en los MOOCs: edad, educación y género. En la *Figura 4* se puede observar que en los extremos de las edades registradas en el *dataset* existe una incidencia que bordea el 100% de probabilidad abandono de los estudiantes; esta probabilidad disminuye y se encuentra una franja de entre 60% y 40% cuando se trata de edades comprendidas entre los 24 y 42 años.

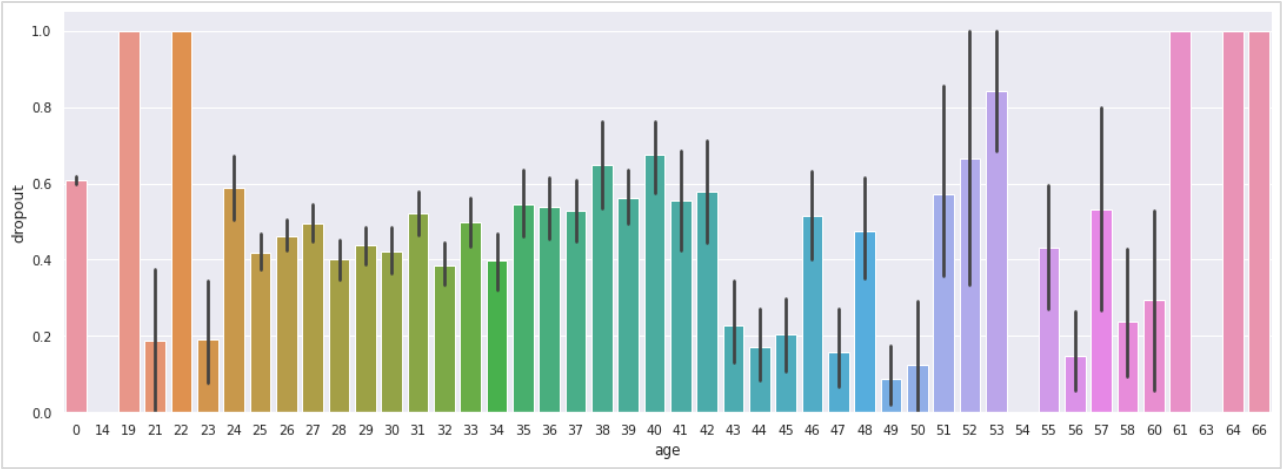

Figura 4: Distribución de porcentaje de abandono por edades

Por otra parte, en la *Figura 5* se puede apreciar el porcentaje de abandono de estudiantes por el nivel de educación que tienen. Los estudiantes de primaria registran un alto porcentaje de abandono de los MOOC, más del 90%. Por otro lado, se tiene los estudiantes de enseñanzas medias, quienes presentan un nivel de abandono bajo, menos del 40%. Finalmente, el resto de estudiantes (master, bachelor, doctorado, etc.) registran un porcentaje de abandono entre el 40 y 60%.

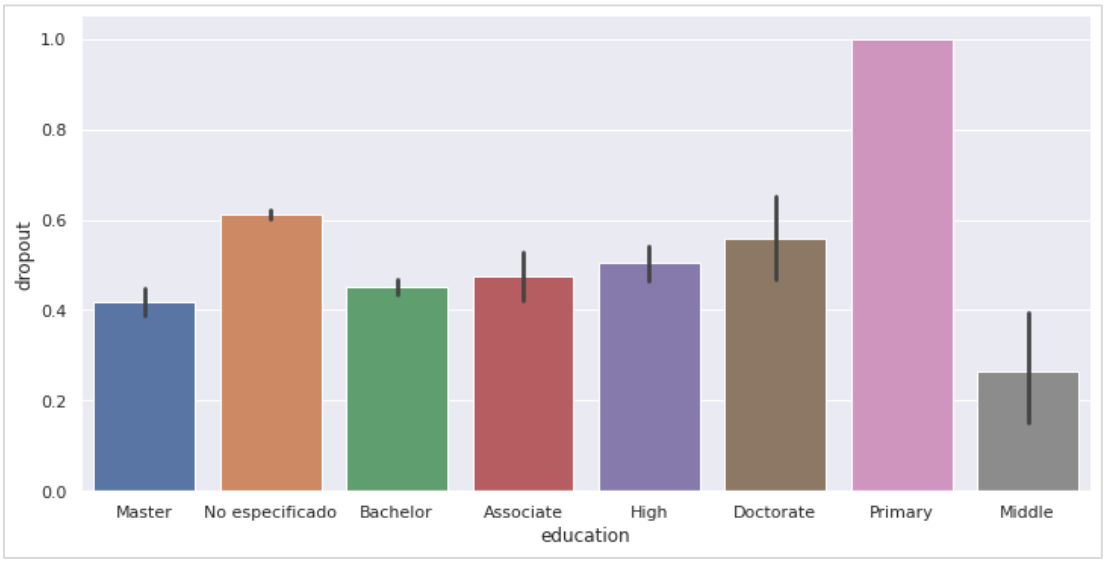

Figura 5: Distribución de porcentaje de abandono por nivel de educación

En la *Figura 6*, podemos apreciar el porcentaje de abandono de estudiantes por género. Se aprecia que la mayor incidencia de abandono se da entre de los estudiantes de género masculino, con un 60%, mientras que las estudiantes de género femenino alcanzan un 47%.

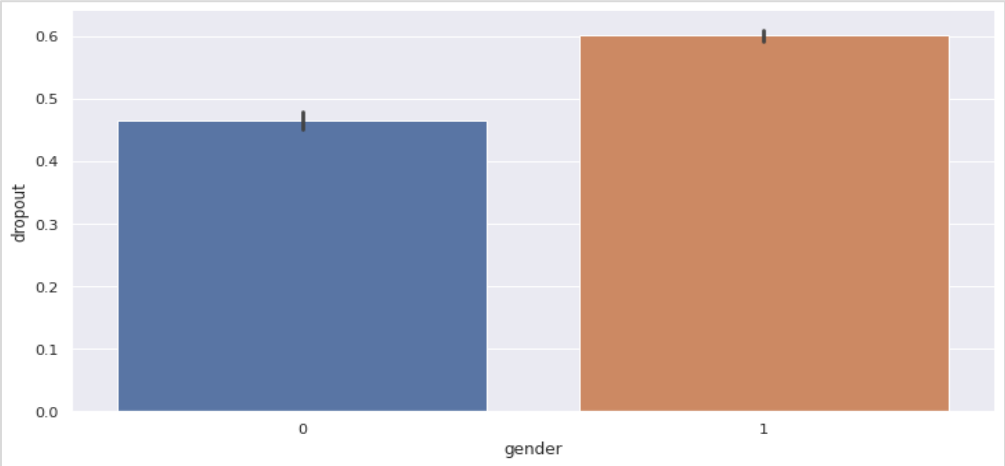

Figura 6: Distribución de porcentaje de abandono por género

En la *Figura 7*, podemos apreciar el porcentaje de abandono por tipo de curso. Se observa que la mayor incidencia de abandono se da en el curso de computación, con un 63%, seguido del curso de economía, con un 53%, mientras que el curso de ingeniería tiene la menor incidencia de abandono, con un 45%.

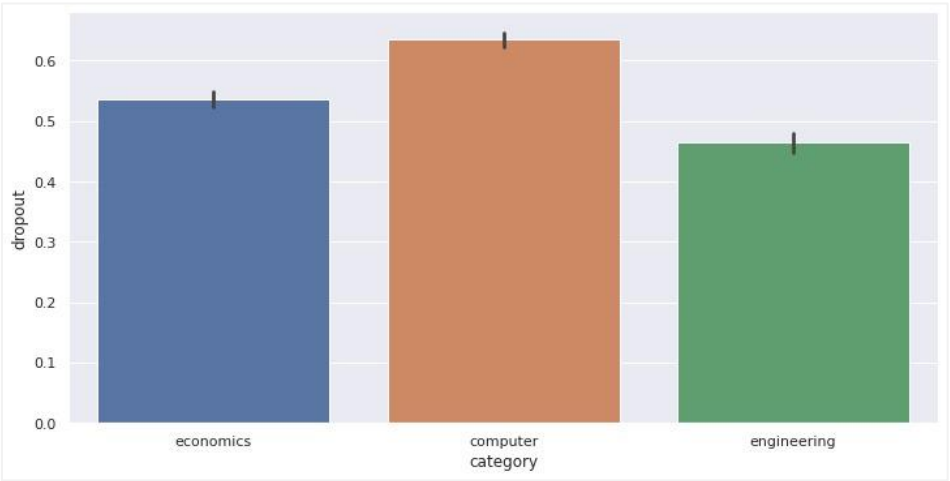

Figura 7: Distribución de porcentaje de abandono por tipo de curso

3.2. Enfoque de los modelos

La necesidad principal de este TFM es brindar una información con la que intentamos predecir el posible abandono de un curso por parte de un estudiante basándose en todas sus actividades registradas en los MOOCs y en sus características (demográficas y otras características). Dichas actividades categorizadas, disponibles en un *dataset,* servirán de insumo para identificar el comportamiento progresivo que va registrando un estudiante durante el desarrollo de un curso. Se pretende que, en caso de identificarse alguna tendencia que señale que estamos ante un posible caso de abandono, sea posible intervenir a tiempo, evitando el abandono y reforzando y potenciando el aprendizaje. Se considera que, en su gran mayoría, los MOOCs seccionan el temario del curso en semanas, y se propone también perfilar a los estudiantes según su actividad: estudiantes registrados (espectadores), estudiantes que inician el curso (auditores) y estudiantes que terminan el curso (actores) [19].

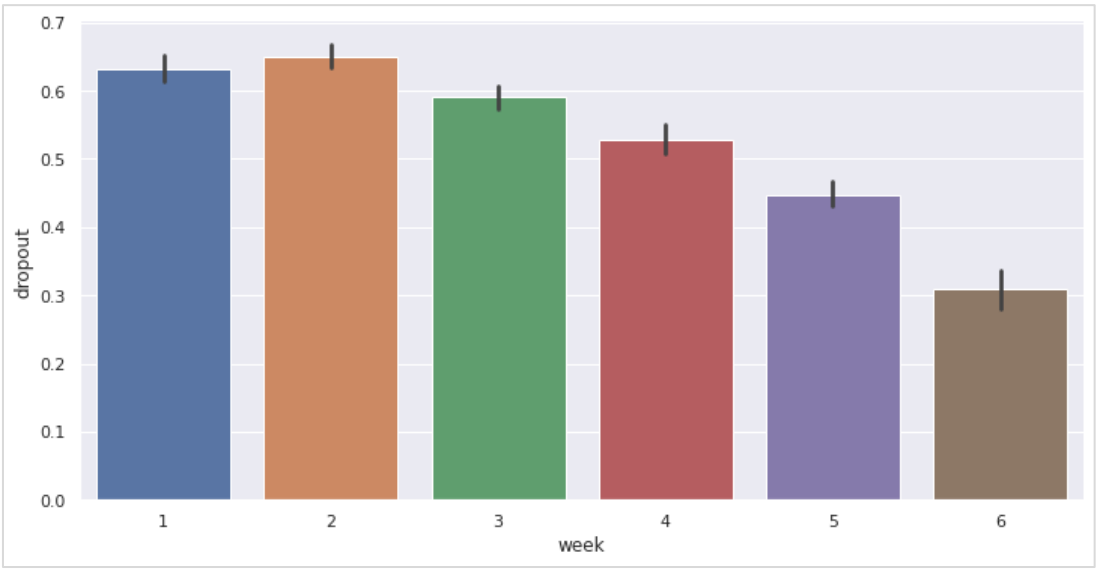

Figura 8: Distribución de abandono de estudiantes por semana

En la *Figura 8* se muestra la distribución de abandono de estudiantes por semana. Se puede apreciar, de manera general, que entre la semana 01 y 03 el porcentaje de abandono bordea el 60%; para las semanas 04 y 05, el porcentaje de abandono se registra entre un 52% y 45%; finalmente, la semana 06, el porcentaje de abandono se registra en un 31%. La incidencia de abandonos va disminuyendo conforme avanzan las semanas, representando el doble de incidencias de abandono al inicio de un curso con respecto a los registros de la última semana.

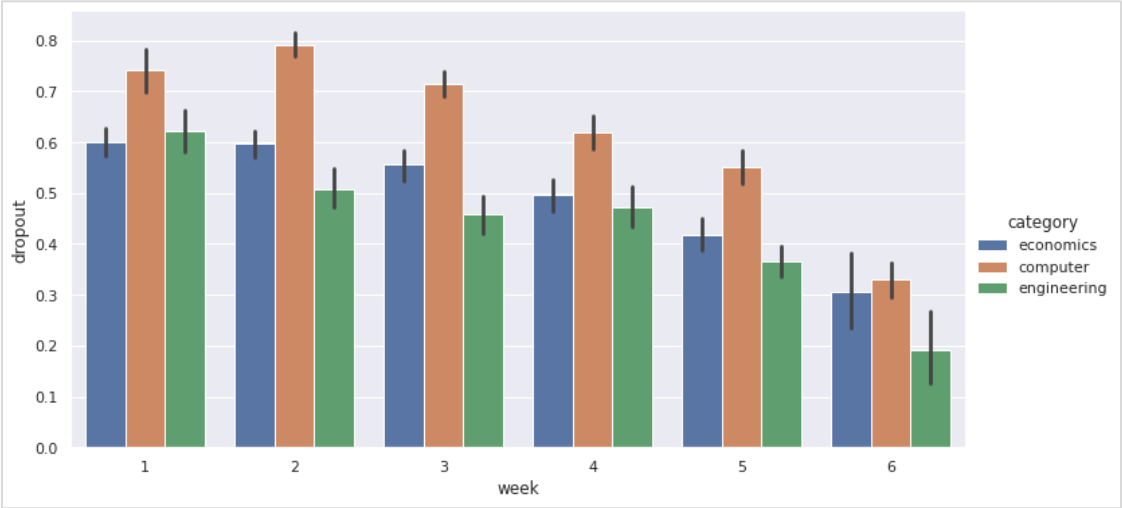

Figura 9: Distribución de abandono de estudiantes por semana / categoría

En la *Figura 9*, se aprecia la distribución de abandono de estudiantes por semana y agrupados por la categoría del curso. Como se puede observar, el curso de computación registra el mayor porcentaje de abandono, bordeando en sus primeras semanas el 80% de abandonos y en su semana final un 42% de abandonos; por otro lado, el curso de ingeniería registra la menor incidencia de abandono, pues registra un 62% en sus primeras semanas y un 19% de abandono en la última semana. Respecto al curso de economía, este registra un porcentaje de abandono que bordea el 60% en las primeras semanas y llega a un 30% de abandono en la última. Por lo tanto, se puede señalar que el curso de computación tiene una mayor incidencia de abandono de estudiantes en sus primeras semanas (80%) respecto a los cursos de economía e ingeniería, obteniendo este último curso el que menor porcentaje de abandono de estudiantes, representado por un 19% en su última semana.

3.3. Tecnología utilizada

La creciente demanda de analizar y transformar los datos en información relevante que nos permita tomar decisiones nos hereda hoy en día muchas herramientas tecnológicas como lenguajes de programación para completar dichas actividades; a continuación, se describen las tecnologías que se han utilizado en el desarrollo de este TFM:

- **Python**

  Es un lenguaje de programación muy útil para analizar los datos extraídos de diferentes fuentes y formatos, pues proporciona librerías y funciones para la extracción y procesamiento de características agrupadas en un conjunto de datos para construir los distintos modelos de aprendizaje automático [20].

- **Google Colab**

  Es un servicio *cloud*, asociado a la suite de Google, que se basa en los *notebooks* de Jupyter, permitiendo la funcionalidad gratuita de GPUs y TPUs, así como un aprovisionamiento de recursos de RAM y HD para realizar nuestras actividades de procesamiento de datos y aprendizaje automático en el desarrollo de este TFM [21].

- **Tensorflow**

  Es una biblioteca de código abierto para aprendizaje automático. Se ha importado dentro de Google colab para facilitar la implementación de algoritmos de DeepLearning [22].

- **Scikit-learn**

  Es una librería para aprendizaje automático de *software* libre para el lenguaje de programación Python, de la cual se han utilizado recursos de algoritmos de regresión logística, KNN, árboles de decisión, SVM, así como métricas para evaluación de los modelos y matriz de confusión [23].

- **Microsoft Word**

  Herramienta de procesamiento de texto para la elaboración de la memoria del TFM [24].

4. Resultados

En esta sección se describen los resultados obtenidos tras la generación de modelos de clasificación supervisada y la generación de un modelo usando redes neuronales. En principio, después de haberse extraído y pre-procesado los datos de los *"log"* de los MOOC, se definen las variables que contienen los registros de interacciones de los estudiantes, así como variables demográficas de los mismos, que pueden influir en el resultado para identificar si un estudiante puede o no abandonar un MOOC. Los resultados alcanzados con los distintos modelos han permitido verificar que en ocasiones es necesario realizar una optimización de las características para obtener mejores resultados.

4.1. Extracción de variables

En principio, se ha realizado la extracción de las variables o características que van a permitir la construcción de los diferentes modelos de clasificación de abandono de estudiantes de los MOOC.

Utilizando librerías de pre procesamiento y análisis de datos que ofrece Python, se han obtenido 2312 registros de estudiantes. Se han definido las siguientes variables:

- **Session_count**
  Define la cantidad de sesiones que ha registrado un estudiante.

- **Action**
  Define la sumatoria de acciones registradas de un estudiante.

- **Age**
  Define la edad del estudiante registrado en el MOOC.

- **Gender**
  Define el género del estudiante (0 Femenino - 1 Masculino).

- **Education**
  Define el nivel de educación declarado por el estudiante al iniciar el MOOC.

- **Seek_video_num**:

  Define la cantidad de búsquedas de vídeos de un estudiante dentro del MOOC.

- **Play_video_num**

  Define la cantidad de reproducciones de vídeos.

- **Pause_video_num**

  Define la cantidad de pausas realizadas en los vídeos.

- **Stop_video_num**

  Define la cantidad de paradas realizadas en los vídeos.

- **Load_video_num**

  Define la cantidad de cargas de vídeos.

- **Problem_get_num**

  Define la cantidad de problemas iniciados sin resolver.

- **Problem_check_num**:

  Define la cantidad de revisiones que hace un estudiante antes de resolver un problema.

- **Problem_save_num**

  Define la cantidad de veces que el estudiante guarda el problema para resolverlo luego.

- **Problem_check_correct_num**

  Define la cantidad de veces que un estudiante ha resuelto correctamente un problema.

- **Problem_check_incorrect_num**

  Define la cantidad de veces que un estudiante ha resuelto un problema de forma incorrecta.

- **Dropout**

  Indica si un estudiante abandona o no el curso MOOC. Esta es la variable que se elige como salida de la clasificación.

4.1.1 Análisis de los valores de las variables

Mediante un análisis de los valores de las variables extraídas, se presentan en las siguientes tablas los atributos contenidos desde un punto de estadística descriptiva.

En primer lugar, en la *Tabla 2* se muestran el número de sesiones y acciones registradas por los estudiantes y los datos demográficos de los mismos:

	Action	***age***	***gender***	***education***	***dropout***
count	*2312.0000*	*2312.0000*	*2312.0000*	*2312.0000*	*2312.0000*
mean	*126.2980*	*6.9952*	*0.2522*	*0.5048*	*0.8097*
std	*519.8838*	*13.3012*	*0.4343*	*1.2179*	*0.3926*
min	*1.0000*	*0.0000*	*0.0000*	*0.0000*	*0.0000*
25%	*4.0000*	*0.0000*	*0.0000*	*0.0000*	*1.0000*
50%	*19.0000*	*0.0000*	*0.0000*	*0.0000*	*1.0000*
75%	*105.0000*	*0.0000*	*1.0000*	*0.0000*	*1.0000*
max	*16454.0000*	*63.0000*	*1.0000*	*7.0000*	*1.0000*

Tabla 2: Descripción de datos (I)

De los datos descritos, se ha identificado que existen un total de 2312 registros. Para las acciones que realiza un estudiante en el curso, se identifica una media de 126 acciones por estudiante. También se puede verificar que el 25% de los estudiantes solo ha registrado 4 acciones, el 50% de ellos ha registrado 19 acciones y el 75% ha registrado 105 acciones. También podemos apreciar que al menos el 75% de los estudiantes registra abandono del curso.

Desde el punto de vista de las acciones registradas que involucran el consumo del contenido en formato de vídeo por parte de los estudiantes, principalmente en la reproducción de los mismos, encontramos que la media de reproducciones de vídeos por estudiante es 23. También se identifica que el 50% de los estudiantes solo registra 4 reproducciones frente a un 25% de estudiantes que registra 20 reproducciones. Esta información se muestra en la *Tabla 3*.

	seek_video_num	*play_video_num*	*pause_video_num*	*stop_video_num*	*load_video_num*
count	*2312.0000*	*2312.0000*	*2312.0000*	*2312.0000*	*2312.0000*
mean	*15.6401*	*22.7764*	*18.7154*	*29.7046*	*13.7111*
std	*40.3492*	*74.5722*	*71.3097*	*445.8088*	*23.6139*
min	*0.0000*	*0.0000*	*0.0000*	*0.0000*	*0.0000*
25%	*0.0000*	*1.0000*	*0.0000*	*0.0000*	*1.0000*
50%	*2.0000*	*4.0000*	*2.0000*	*0.0000*	*4.0000*
75%	*12.0000*	*20.0000*	*15.0000*	*4.0000*	*15.2500*
max	*475.0000*	*2122.0000*	*2113.0000*	*15415.0000*	*258.0000*

Tabla 3: Descripción de datos (II)

Desde el punto de vista de acciones registradas que involucran la realización de actividades de evaluación por parte de los estudiantes (ver *Tabla 4*), se detecta que existe poca actividad, pues sólo un 25% de los estudiantes registra interacción en este tipo de actividades.

	problem_get_num	*problem_check_num*	*problem_save_num*	*problem_check_correct_num*	*problem_check_incorrect_num*
count	*2312.0000*	*2312.0000*	*2312.0000*	*2312.0000*	*2312.0000*
Mean	*11.5921*	*6.9165*	*0.1851*	*3.7383*	*3.1453*
Std	*35.4273*	*16.6880*	*0.6889*	*8.3682*	*9.2478*
Min	*0.0000*	*0.0000*	*0.0000*	*0.0000*	*0.0000*
25%	*0.0000*	*0.0000*	*0.0000*	*0.0000*	*0.0000*
50%	*0.0000*	*0.0000*	*0.0000*	*0.0000*	*0.0000*
75%	*7.0000*	*6.0000*	*0.0000*	*4.0000*	*2.0000*
Max	*421.0000*	*176.0000*	*9.0000*	*70.0000*	*133.0000*

Tabla 4: Descripción de datos (III)

Figura 10: Mapa de calor de correlación de variables respecto al DropOuts

4.2. Clasificación supervisada

La clasificación como método de aprendizaje supervisado tiene como objetivo clasificar en grupos atendiendo a datos previamente etiquetados.

Durante el desarrollo de este TFM se busca identificar patrones en los estudiantes que permitan clasificarlos en 2 grupos el grupo que abandona un MOOC y otro que se mantiene en él. Para lograr dicho objetivo es necesario introducir datos previamente etiquetados para obtener una salida, generada por los diferentes algoritmos de clasificación supervisada [25].

4.2.1 K-Nearest Neighbort

Dentro de los algoritmos de clasificación seleccionados, el algoritmo de vecinos próximos K-NN (K-Nearest Neighbort) se caracteriza por clasificar cada caso nuevo que detecte considerando la clase más frecuente a la que pertenecen sus vecinos más cercanos [26].

		X_1	...	X_j	...	X_n	C
$(\mathbf{x}_1, c_1)$	1	x_{11}	...	x_{1j}	...	x_{1n}	c_1
	⋮	⋮		⋮		⋮	⋮
$(\mathbf{x}_i, c_i)$	i	x_{i1}	...	x_{ij}	...	x_{in}	c_i
	⋮	⋮		⋮		⋮	⋮
$(\mathbf{x}_N, c_N)$	N	x_{N1}	...	x_{Nj}	...	x_{Nn}	c_N
$\mathbf{x}$	$N+1$	$x_{N+1,1}$	...	$x_{N+1,j}$	...	$X_{N+1,n}$	?

Figura 11: Notación principal de algoritmo de K-NN

Mediante el uso del algoritmo de vecinos más cercanos (KNN), se pretende obtener el número de vecinos óptimos para poder determinar si un estudiante abandona o no un MOOC.

En ese sentido, se ha procedido a realizar un modelo de clasificación inicial que contempla 1 solo vecino, es decir n_neighbors=1.

```
KNeighborsClassifier(algorithm='auto', leaf_size=30,
metric='minkowski', metric_params=None, n_jobs=None, n_neighbors=1,p=2,
weights='uniform')
```

Bloque de código 1: Modelo de clasificación inicial (n_neighbobrs=1)

Con este modelo se obtienen los siguientes resultados:

```
              precision    recall  f1-score   support

           0       0.41      0.35      0.38       135
           1       0.85      0.88      0.86       559

    accuracy                           0.78       694
   macro avg       0.63      0.61      0.62       694
weighted avg       0.76      0.78      0.77       694

0.776657060518732
```

Bloque de código 2: Resultados del modelo de clasificación inicial

Según el reporte de clasificación, nuestro clasificador de KNN obtiene una precisión del 85% de los casos que un estudiante abandone un MOOC, frente a un 41% para los casos en los que el estudiante no abandone un MOOC. Así mismo, se observa que nuestro modelo alcanza un 77,66% de exactitud.

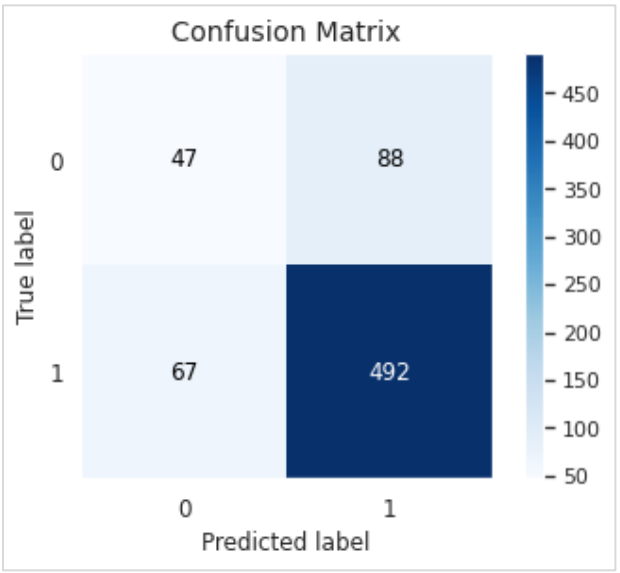

Figura 12: Matriz de confusión 1 del modelo K-NN

Así mismo, se identifica en la matriz de confusión (ver *Figura 12*) que nuestro modelo ha clasificado correctamente 492 casos de abandono y 67 casos de manera incorrecta. Por otro lado, se aprecia que al intentar predecir casos en los que el estudiante no abandona el MOOC, solo acierta en 47 casos frente a 87 en los que clasifica de manera incorrecta.

Dados los resultados iniciales, se identifica que es necesario encontrar cuál es el mejor parámetro para n_neighbors que nos permita alcanzar mejores resultados. Para ello se ha iterado con valores que van de 1 a 30 n_neighbors, con el fin de identificar la menor tasa de error obtenida y probar el modelo nuevamente.

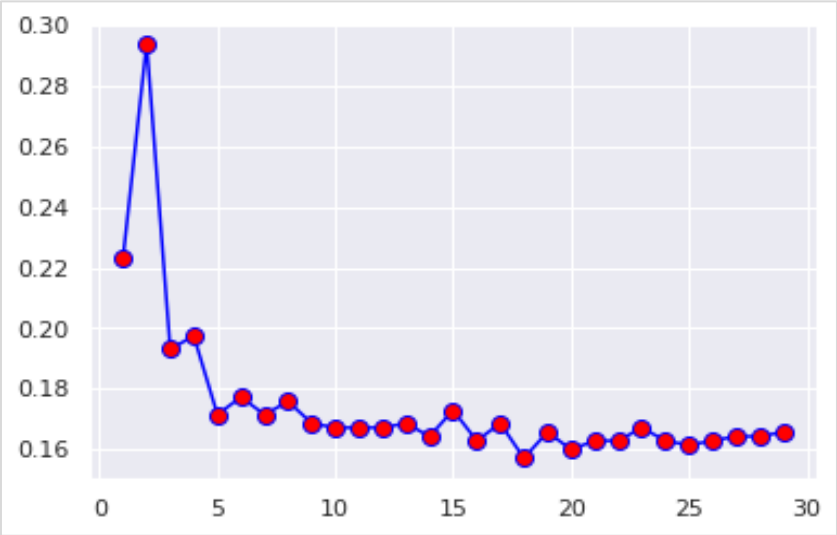

Figura 13: Identificación de parámetro de n_neighbors

En la *Figura 13* apreciamos que la menor tasa de error es **0.15**, la cual señala que el número n_neighbors es **18**. Por lo que se procede a configurar dicho parámetro con ese valor en el modelo, con el cual se obtienen los siguientes resultados.

```
              precision    recall  f1-score   support

           0       0.70      0.34      0.46       135
           1       0.86      0.96      0.91       559

    accuracy                           0.84       694
   macro avg       0.78      0.65      0.68       694
weighted avg       0.83      0.84      0.82       694

0.8429394812680115
```

Bloque de Código 3: K-NN (n_neighbors = 18)

Tras la optimización de parámetro n_neighbors obtenemos una precisión del 86% de los casos que un estudiante abandone un MOOC, frente a un 70% para los casos en los que el estudiante no abandone un MOOC. Así mismo, se observa que este modelo alcanza un 84,29% de exactitud.

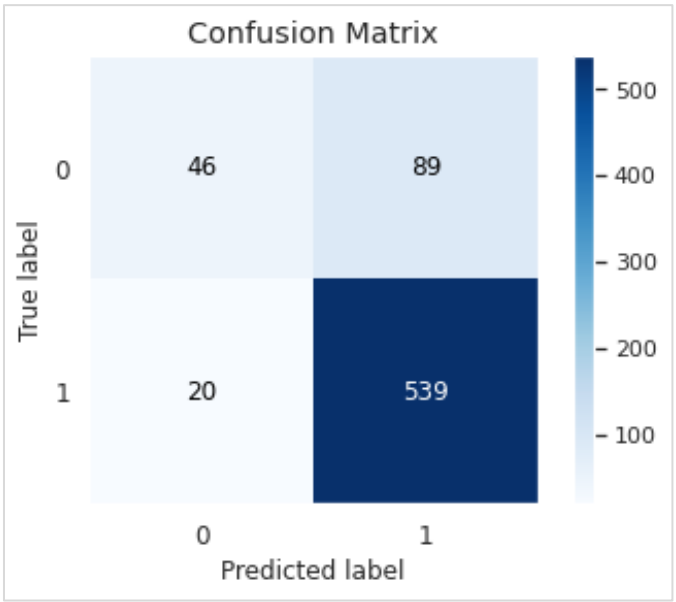

Figura 14: Matriz de confusión 2 del modelo K-NN

Así mismo, se identifica en la matriz de confusión que nuestro modelo ha clasificado 539 casos de abandono de manera correcta y 20 casos de manera incorrecta. Por otro lado, se aprecia que al intentar predecir casos en los que el estudiante no abandona el MOOC, solo acierta en 46 casos frente a 89 que clasifica de manera incorrecta, lo que puede significar que requiere una mayor cantidad de información para entrenar mejor la clasificación de no abandono.

4.2.2 Árboles de decisión

Dentro de los algoritmos de clasificación seleccionados, el algoritmo de árboles de decisión se caracteriza por basarse en un particionamiento recurso del dominio de las variables predictoras, mediante el cual se va a poder representar el conocimiento del problema por medio de la estructura de un árbol [27].

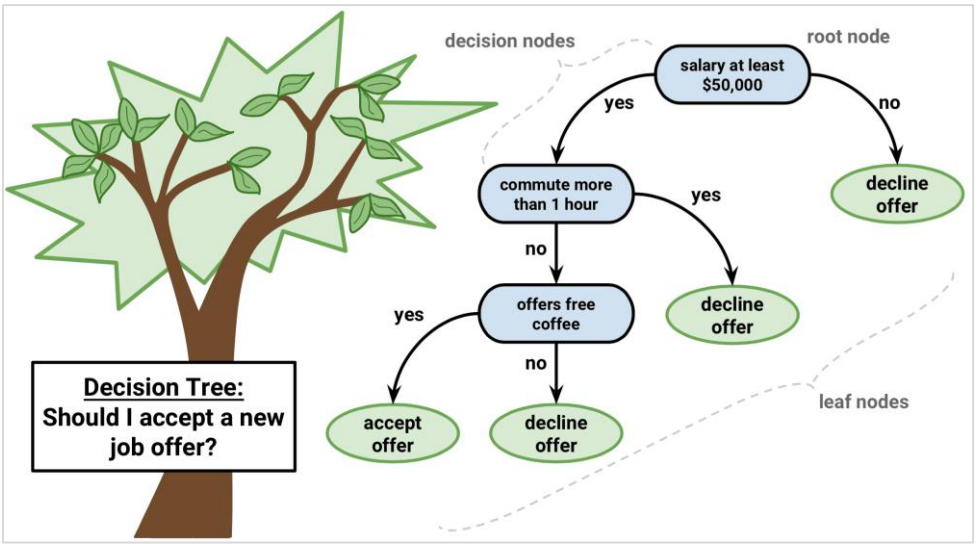

Figura 15: Diagrama del árbol [27]

Componentes de un árbol de decisión:

- Nodo raíz: población completa o muestra.
- Ramificación.
- Nodo de decisión.
- Nodo terminal y hoja
- Poda.
- Rama/sub-árbol.
- Nodos padre e hijo.

Para la construcción de este modelo se ha utilizado el algoritmo de árboles de decisión de tipo clasificador, cuyo valor en el nodo terminal se reduce a la moda de las observaciones del conjunto de entrenamiento que han “caído” en esa región.

En principio se instancia el clasificador y se entrena el modelo con los parámetros por defecto:

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None,
criterion='gini', max_depth=None, max_features=None,
max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, presort='deprecated',
random_state=None, splitter='best')
```

Bloque de Código 4: Árboles de decisión tipo clasificador

Obteniéndose los siguientes resultados:

```
              precision    recall  f1-score   support

           0       0.39      0.38      0.38       135
           1       0.85      0.86      0.85       559

    accuracy                           0.76       694
   macro avg       0.62      0.62      0.62       694
weighted avg       0.76      0.76      0.76       694

0.7636887608069164
```

Bloque de Código 5: Resultado del Árbol de Clasificación

Según el reporte de clasificación, nuestro clasificador de Árbol de Clasificación obtiene una precisión del 85% de los casos que un estudiante abandone un MOOC, frente a un 39% para los casos en los que el estudiante no abandone un MOOC. Así mismo, se observa que nuestro modelo alcanza un 76,66% de exactitud.

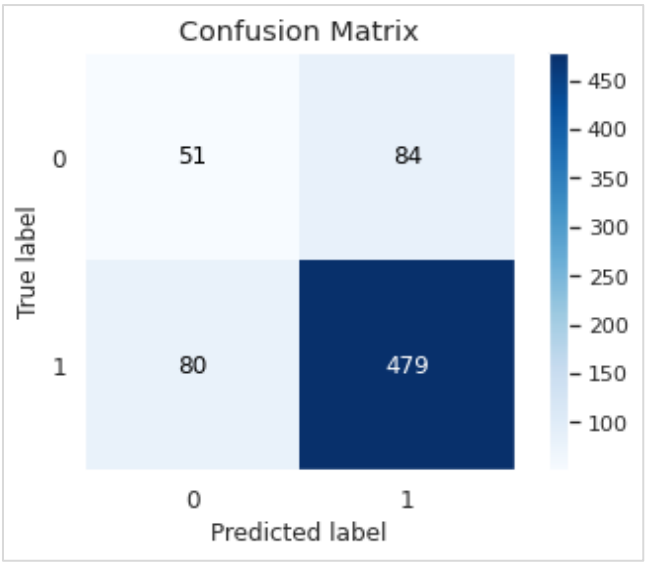

Figura 16: Matriz de confusión 1 del modelo de árboles de decisiones

Así mismo, se identifica en la matriz de confusión que este modelo ha clasificado correctamente 479 casos de abandono y 80 casos de manera incorrecta. Por otro lado, se aprecia que al intentar predecir casos en los que el estudiante no abandona el MOOC, solo acierta en 51 casos frente a 84, en los que clasifica de manera incorrecta.

Habiendo mencionado que este modelo se ha entrenado con los parámetros del árbol de clasificación por defecto, se realiza la búsqueda de los mejores parámetros usando GridSearchCV, de donde se obtienen los siguientes parámetros:

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None,
criterion='gini',       max_depth=2, max_features=None, max_leaf_nodes=None
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, presort='deprecated',
                        random_state=None, splitter='best')
```

Bloque de Código 6: Uso del GridSearchCV

Los mismos que permiten instanciar el clasificador con los nuevos parámetros y ajustar los datos de entrenamiento de nuestro modelo, para obtener los siguientes resultados:

```
              precision    recall  f1-score   support

           0       0.76      0.21      0.33       135
           1       0.84      0.98      0.90       559

    accuracy                           0.83       694
   macro avg       0.80      0.60      0.62       694
weighted avg       0.82      0.83      0.79       694

0.8328530259365994
```

Bloque de Código 7: Resultado del uso del GridSearchCV

De la optimización de parámetro usando GridSearchCV, obtenemos una precisión del 84% de los casos que un estudiante abandone un MOOC, frente a un 76% para los casos en los que el estudiante no abandone un MOOC. Así mismo, se observa que este modelo alcanza un 83,28% de exactitud.

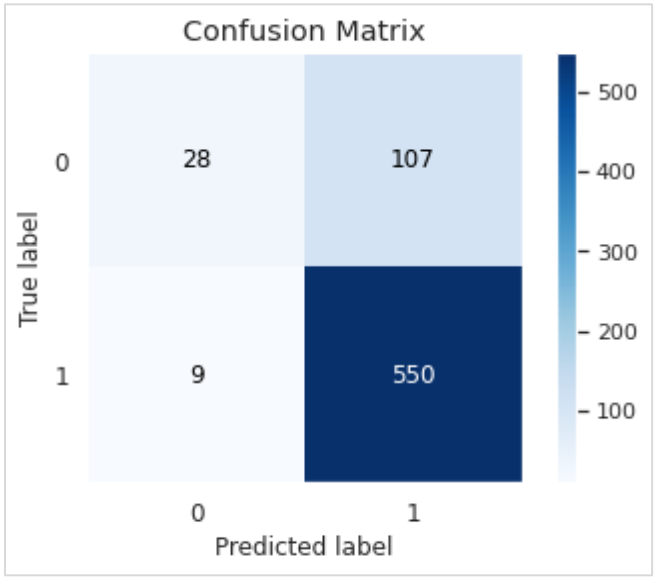

Figura 17: Matriz de confusión 2 del modelo de árboles de decisiones

Así mismo, se identifica en la matriz de confusión que este modelo ha clasificado correctamente 550 casos de abandono y 20 casos de manera incorrecta. Por otro lado, se aprecia que al intentar predecir casos en los que el estudiante no abandona el MOOC, solo acierta en 28 casos frente a 107 que clasifica de manera incorrecta.

4.2.3 Random Forest

Conformado por un conjunto de árboles de decisión combinados de forma que ven distintas porciones de los datos, permitiendo que cada árbol se entrene con distintas muestras de datos para un mismo problema, compensando unos errores con otros, tenemos una predicción que generaliza mejor [28].

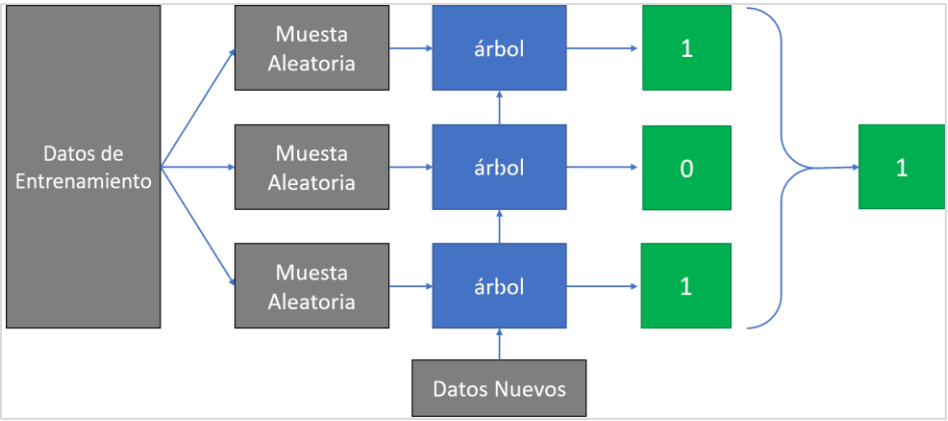

Figura 18: Árbol de decisión

La implementación realizada corresponde a la de un RandomForestClassifier, para el cual se definen algunos hiper-parámetros:

Propios del Bosque Aleatorio:

- **n_estimators**

  Número de árboles que va a tener el bosque aleatorio. Normalmente cuantos más mejor, pero a partir de cierto punto deja de mejorar y sólo hace que vaya más lento. Un buen valor por defecto puede ser el uso de 100 árboles.

- **n_jobs**

  Número de *cores* que se pueden usar para entrenar los árboles. Cada árbol es independiente del resto, así que entrenar un bosque aleatorio es una tarea muy paralelizable. Por defecto sólo utiliza 1 *core* de la CPU. Para mejorar el rendimiento se pueden usar tantos *cores* como se estimen necesarios. Si se usan n_jobs = -1, se está indicando que se quieren usar tantos *cores* como tenga la máquina.

- **max_features**

  Usa forma de garantizar que los árboles son diferentes, es que todos se entrenan con una muestra aleatoria de los datos. Si queremos que todavía sean más diferentes, podemos hacer que distintos árboles usen distintos atributos. Esto puede ser útil especialmente cuando algunos atributos están relacionados entre sí.

Regularización (también disponibles para árboles de decisión):

- **max_depth**

  Profundidad máxima del árbol.

- **min_samples_split**

  Número mínimo de muestras necesarias antes de dividir este nodo. También se puede expresar en porcentaje.

- **min_samples_leaf**

  Número mínimo de muestras que debe haber en un nodo final (hoja). También se puede expresar en porcentaje.

- **max_leaf_nodes**

  Número máximo de nodos finales.

Para el modelo construido utilizando el algoritmo de Random Forest, se ha identificado que se obtienen mejores resultados, utilizando los datos sin normalizar, pero realizando GridSearchCV.

```
param_grid = {
    'n_estimators': [200, 500],
    'max_features': ['auto', 'sqrt', 'log2'],
    'max_depth' : [4,5,6,7,8],
    'criterion' :['gini', 'entropy']
}
```

Bloque de Código 8: Datos sin normalizar con GridSearchCV

Se definen los parámetros para optimizar el clasificador inicial, para el GridSearch utilizando validación cruzada (CV), para luego ajustar los resultados de la búsqueda con los datos de entrenamiento.

```
{'criterion': 'gini',
 'max_depth': 4,
 'max_features': 'auto',
 'n_estimators': 200}
```

Bloque de Código 9: Optimizando el clasificador inicial

Obteniendo como resultado los datos presentados en el bloque de código 03, los mismos que son definidos en una nueva versión del clasificador y ajustados con los nuevos datos de entrenamiento, bajo los siguientes parámetros:

```
rfc1=RandomForestClassifier(random_state=42, max_features='log2', n_estima
max_depth=6, criterion='gini')

rfc1.fit(x_train, y_train)

RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=6, max_features='log2',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=500,
                       n_jobs=None, oob_score=False, random_state=42,
                       verbose=0, warm_start=False)
```

Bloque de Código 10: Ajustes a los parámetros

Verificamos nuestro reporte de clasificación:

```
              precision    recall  f1-score   support

           0       0.67      0.33      0.45        99
           1       0.86      0.96      0.91       422

    accuracy                           0.84       521
   macro avg       0.77      0.65      0.68       521
weighted avg       0.82      0.84      0.82       521

0.8426103646833013
```

Bloque de Código 11: Validación del reporte de clasificación

De la optimización de parámetro usando GridSearchCV, obtenemos una precisión del 86% de los casos que un estudiante abandone un MOOC, frente a un 67% para los casos en los que el estudiante no abandone un MOOC. Así mismo, se observa que este modelo alcanza un 84,26% de exactitud.

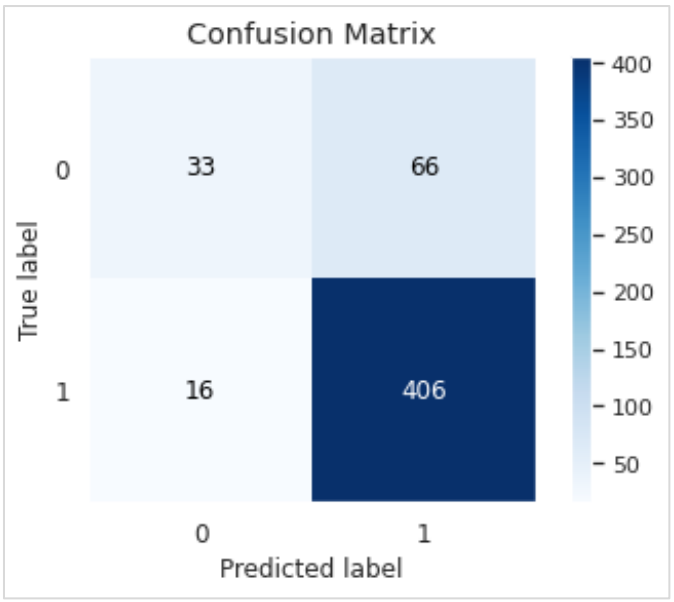

Figura 19: Matriz de confusión del modelo Random Forest

Así mismo, se identifica en la matriz de confusión que nuestro modelo ha clasificado 406 casos de abandono de manera correcta y 16 casos de manera incorrecta. Por otro lado, se aprecia que al intentar predecir casos en los que el estudiante no abandona el MOOC, solo acierta en 33 casos frente a 66 que clasifica de manera incorrecta.

4.2.4 Vectores de máquinas de soporte (SVM)

Es un algoritmo de clasificación capaz de entrenarse a partir de una serie de elementos para posteriormente predecir a cuál de las posibles clases pertenecen nuevos elementos.

Los elementos que se proporcionan al algoritmo tanto para su entrenamiento como para la predicción, vienen definidos por una serie valores que representan mediciones realizadas sobre diferentes características del mismo (por ejemplo, su peso, altura, anchura, color, etc.). Al definir cada elemento usando "n" atributos, realmente estamos representando puntos en un espacio n-dimensional.

El algoritmo SVM es capaz de crear un hiperplano en este espacio n-dimensional que sirve de frontera entre ambas clases, de forma que todos los elementos que queden a un lado de los hiperplanos pertenecerán a una clase y los que queden al otro lado del hiperplano pertenecerán a la otra clase [29].
Matemáticamente, el hiperplano se define como:

$$W^T x_i + b = 0$$

Siendo:

- **W** el vector ortogonal al hiperplano.
- **b** el coeficiente de intersección.

Aquellos elementos que den un resultado positivo serán considerados de una clase y los que den un resultado negativo, de la otra. Por tanto, la función de clasificación será:

$$f(x) = sgn(W^T x + b)$$

Para el caso de SVM, ha sido necesario normalizar los datos y realizar una búsqueda de parámetros óptimos mediante GridSearch la misma que nos ha arrojado los mejores resultados.

```
param_grid = {'C': [0.1, 1, 10, 100, 1000],
              'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
              'kernel': ['rbf']}
 grid = GridSearchCV(SVC(), param_grid, refit = True, verbose = 4)
 grid.fit(x_train_norm, y_train_norm)
```

Bloque de Código 12: Normalización de datos y búsqueda mediante GridSearch

Se establecen los parámetros iniciales para la búsqueda de nuestros mejores parámetros con el método GridSearch, ajustando nuestros datos a los valores del mismo, donde podemos identificar que nuestros mejores parámetros son:

```
{'C': 1, 'gamma': 1, 'kernel': 'rbf'}
```

Bloque de Código 13: Establecimiento de parámetros iniciales en búsqueda de mejores parámetros

Obteniendo como *best estimator*, los siguientes parámetros:

```
SVC(C=1, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=1, kernel='rbf',
max_iter=-1, probability=False, random_state=None, shrinking=True,
tol=0.001,verbose=False)
```

Bloque de Código 14: Identificación de mejores parámetros

Verificamos nuestros reportes de clasificación del modelo SVM:

```
              precision    recall  f1-score   support

         0.0       0.76      0.20      0.32       140
         1.0       0.83      0.98      0.90       554

    accuracy                           0.83       694
   macro avg       0.79      0.59      0.61       694
weighted avg       0.81      0.83      0.78       694

0.8256484149855908
```

Bloque de Código 15: Validación del reporte de clasificación

De la optimización de parámetro usando GridSearchCV [30], obtenemos una precisión del 83% de los casos que un estudiante abandone un MOOC, frente a un 76% para los casos en los que el estudiante no abandone un MOOC. Así mismo, se observa que nuestro modelo alcanza un 82,56% de precisión.

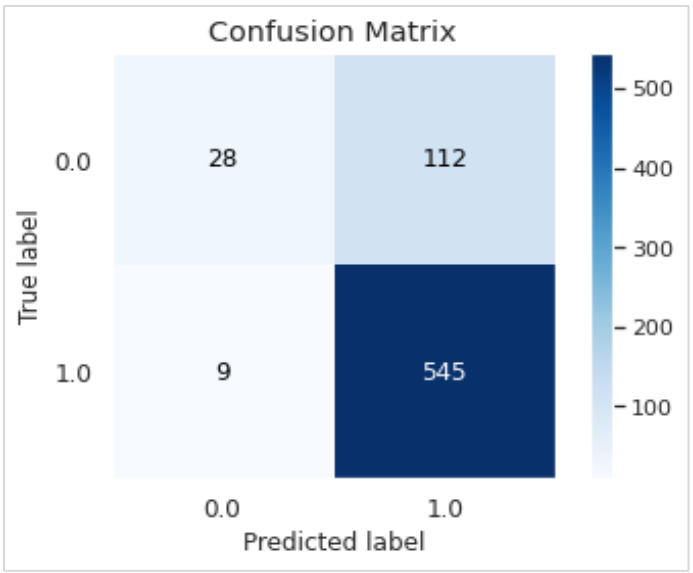

Figura 20: Matriz de confusión del modelo SVM

Así mismo, se identifica en la matriz de confusión que nuestro modelo ha clasificado 545 casos de abandono de manera correcta y 9 casos de manera incorrecta. Por otro lado, se aprecia que al intentar predecir casos en los que el estudiante no abandona el MOOC, solo acierta en 28 casos frente a 112 que clasifica de manera incorrecta.

4.2.5 Artificial Neural Network (ANN)

Mediante el uso de redes neuronales, se ha utilizado el módulo de keras para la construcción de un modelo secuencial, que nos permita clasificar el abandono o no, de un estudiante de MOOC [18].

```
model = Sequential()
model.add(Dense(12, input_dim=15, activation='relu'))
model.add(Dense(15, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
```

Bloque de código 16: Modelo secuencial

Se instancia la secuencialidad de un modelo, y se establecen los parámetros para compilar el modelo secuencial.

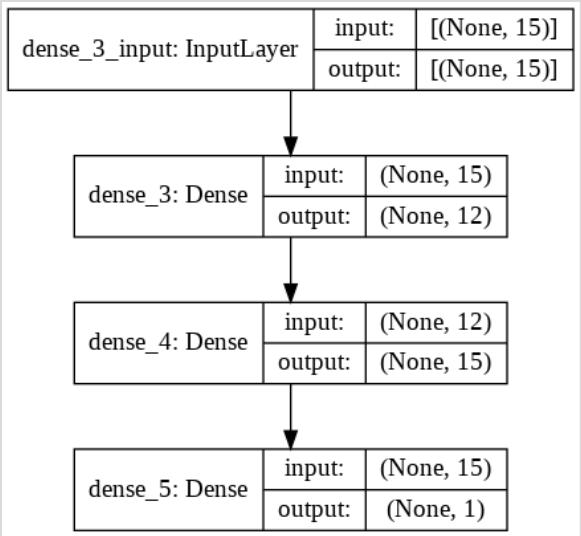

Figura 21: Resultado de la arquitectura de la red neuronal

Se imprime el resultado de nuestra arquitectura de la red neuronal para identificar el número de entradas, capas intermedias y salidas.

```
model.fit(x_, y_, epochs=50, batch_size=10, validation_split=0.3)
```

Bloque de código 17: Ajuste del modelo con parámetros deseados

Luego de haber definido nuestra red neuronal, ajustamos nuestro modelo con los parámetros deseados.

```
              precision    recall  f1-score   support

         0.0       0.81      0.39      0.53       440
         1.0       0.87      0.98      0.92      1872

    accuracy                           0.87      2312
   macro avg       0.84      0.68      0.72      2312
weighted avg       0.86      0.87      0.85      2312

0.8663494809688581
```

Bloque de código 18: Reporte de clasificación (% estudiante abandone un MOOC)

Nuestro reporte de clasificación nos muestra una precisión del 87% de los casos en que un estudiante abandone un MOOC, frente a un 81% para los casos en los que el estudiante no abandone un MOOC. Así mismo, se observa que nuestro modelo alcanza un 86,63% de exactitud.

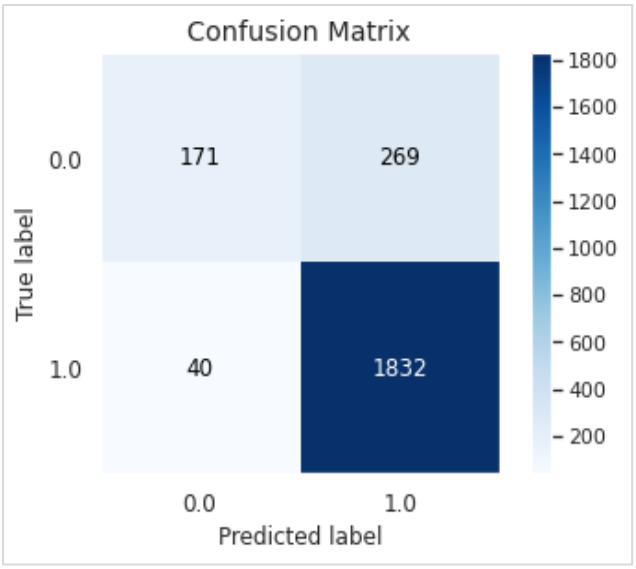

Figura 22: Matriz de confusión - Casos que un estudiante abandone un MOOC

Así mismo, se identifica en la matriz de confusión que este modelo ha clasificado 1832 casos de abandono de manera correcta y 40 casos de manera incorrecta. Por otro lado, se aprecia que al intentar predecir casos en los que el estudiante no abandona el MOOC, solo acierta en 171 casos frente a 269, que clasifica de manera incorrecta.

4.3. Comparación de los resultados

Respecto a los resultados obtenidos con los diferentes algoritmos utilizados para construir nuestros modelos de clasificación (ver *Tabla 5*), podemos decir que, nuestro mejor porcentaje de clasificación ha sido alcanzado por el modelo de red neuronal secuencial con un 86%. Cabe mencionar que, la red neuronal ha generado una mayor cantidad de datos dentro de su capa oculta, en comparación a la cantidad de datos con los que se han entrenado el resto de modelos, lo que muestra un ligero incremento en la tasa de aciertos.

	K-NN	Decision Tree	Random Forest	SMV	ANN
Tasa de Acierto	84.29%	83.28%	84.26%	82.56%	86.63%

Tabla 5: Tasa de acierto (%)

La tasa de acierto también nos permite evidenciar que los otros modelos se encuentran dentro de un rango 82.56% y 84.29%, lo cual demuestra que son resultados muy cercanos pese a ser modelos diferentes, lo que nos lleva a reflexionar sobre la calidad de los datos y su importancia en la obtención de este tipo de resultados.

Por otro lado, se debe mencionar que los modelos de clasificación supervisada han requerido de una búsqueda de hiper parámetros para optimizar sus tasas de aciertos.

5. Conclusiones y trabajo futuro

5.1. Conclusiones

Durante el trabajo desarrollado se han revisado y estudiado conceptos relacionados con Analíticas del aprendizaje (Learning Analytics), que se presentan como una agrupación de técnicas y herramientas que buscan mejorar el proceso de aprendizaje de un estudiante y su entorno. Para este TFM, los componentes relacionados con la interacción generada por los estudiantes en el MOOC, así como sus registros demográficos, son insumos valiosos para realizar un proceso de analítica en el cual se ha explorado la posibilidad de identificar la incidencia de abandono de un estudiante registrado en un MOOC utilizando tres tipos de cursos y escogiendo el que registra mayores niveles de abandono para en adelante utilizar dicha información.

El proceso de extracción de las características y preprocesamiento de los datos con los que se ha trabajado ha significado un reto importante en el proceso de aprendizaje sobre técnicas de análisis exploratorio de datos, preprocesamiento de datos, visualización, construcción y mejoramiento de modelos e interpretación de resultados.

Finalmente, la construcción de modelos propios basados en algoritmos de aprendizaje supervisado y agregando un modelo de redes neuronales, ha arrojado que, en base a los datos con los que se han entrenado dichos modelos, existe mucha similitud en los resultados alcanzados.

Por lo que podemos concluir que la aplicación de estos algoritmos y los resultados obtenidos se encuentran muy relacionados con la calidad y cantidad de los datos asociadas a la adecuada búsqueda e identificación de parámetros para los modelos, que permitirán a los actores del proceso de aprendizaje que utilicen Learning Analytics, consultar dicha información para la toma de decisiones que generen mejoras ya sea de forma preventiva o correctiva en beneficio del proceso de aprendizaje de futuros estudiantes de los MOOC.

5.2. Trabajo futuro

Considerando los resultados alcanzados y descritos en esta memoria, debemos considerar la posibilidad de incluir una pequeña encuesta previo al inicio del MOOC para recabar más información de los estudiantes, con el fin de identificar otros aspectos como, por ejemplo, cuáles son las condiciones del ambiente donde desarrollará el curso (es decir, si cuenta con un espacio adecuado, un estudio aislado de ruidos o está en la sala de su casa sometido a factores de distracción) o las características del dispositivo desde el que desarrolla el curso, entre otros posibles factores, que nos permitan perfilar de mejor manera a los estudiantes y complementar la información registrada en los logs, la cual será nuestro insumo para construir futuros modelos.

En segundo lugar, el hecho de haber explorado algoritmos de aprendizaje supervisado y redes neuronales no limita la posibilidad de explorar la construcción de modelos basados en algoritmos de aprendizaje no supervisado.

Por otro lado, existiría la posibilidad de clasificar las interacciones por grupo de actividades de modo que, basándose en registros históricos, se puedan transmitir mensajes como, por ejemplo, cuál es el momento clave para entender un tópico en un vídeo formativo, y alertar al estudiante que debe prestar mayor atención. Estas y otras alertas podrían ser enviadas a través de mensajería instantánea en base a categorías de estudiantes previamente establecidas con datos históricos.

Sin embargo, el simple hecho de clasificar a los estudiantes no garantiza un óptimo aprovechamiento de los técnicas y herramientas de *learning analytics*, sino que estos deben estar acompañados del *feedback* generado por los responsables del MOOC, principalmente los profesores, pues son ellos quienes pueden y deben valorar finalmente el impacto que genera la distribución del contenido del curso en los estudiantes, utilizando información recogida automáticamente y complementándola con la información disponible en fuentes como foros de preguntas y respuestas. Una vez identificados estos aspectos, el profesor podría reflexionar sobre el diseño del MOOC y las mejoras que se podrían implementar para mejorar y garantizar la finalización del curso.

Finalmente, la implementación de un sistema integrado de *Learning Analytics* que se integre y recoja la información de los estudiantes, procese datos y proponga acciones concretas a tomar por parte de los actores involucrados en el proceso de aprendizaje sería una interesante línea de trabajo futura.

Referencias

[1] S. Knight, S. Buckinnghan, “Theory and Learning Analytics”, Handbook of Learning Analytics. First Edition. Solar Society for Learning Analytics Research, pp. 17-22, Marzo 2017.

[2] C. Anderson, “The End of Theory: The Data Deluge Makes the Scientific Method Obsolete”, *Wired Magazine*, Vol. 16, no. 07, pp. 1-3, Junio 2008.

[3] C. Jin (2020, Agosto), MOOC student dropout prediction model based on learning behavior features and parameter optimization. [En línea]. Disponible en: https://www.researchgate.net/publication/343615890_MOOC_student_dropout_prediction_model_based_on_learning_behavior_features_and_parameter_optimization

[4] X. Lu, S. Wang, J. Huang, W. Chen, Z. Yan (2017, Marzo), What Decides the Dropout in MOOCs? [En línea]. Disponible en: https://www.researchgate.net/publication/315469257_What_Decides_the_Dropout_in_MOOCs

[5] C. Brooks, C. Thompson, “Predictive Modelling in Teaching and Learning”, Handbook of Learning Analytics. First Edition. Solar Society for Learning Analytics Research, pp. 61-68, Marzo 2017.

[6] Detra D. Johnson, “The Origins of MOOCs: The Beginning of the Revolution of All at Once-Ness”, Ph.D. Student, Texas A&M University, 2014.

[7] MoocLab (2017). [En línea]. Disponible en: https://www.mooclab.club/resources/mooclab-report-the-global-mooc-landscape-2017.214

[8] Coursera. [En línea]. Disponible en: https://es.coursera.org/

[9] edX. [En línea]. Disponible en: https://www.edx.org/es

[10] Future Learn. [En línea]. Disponible en: https://www.futurelearn.com/

[11] XueTangX. [En línea]. Disponible en: https://www.xuetangx.com/global

[12] Información actualizada sobre los productos y servicios de XueTangX. [En línea]. Disponible en: https://en.wikipedia.org/wiki/XuetangX

[13] Solar Society for Learning Analytics Research. [En línea]. Disponible en: https://www.solaresearch.org/about/what-is-learning-analytics/

[14] U. Hoppe, “Computational Methods for the Analysis of Learning and Knowledge Building Communities”, Handbook of Learning Analytics. First Edition. Solar Society for Learning Analytics Research, pp. 23-33, Marzo 2017.

[15] F. Sciarrone, "Machine Learning and Learning Analytics: Integrating Data with Learning," 2018 17th International Conference on Information Technology Based Higher Education and Training (ITHET), Olhao, 2018, pp. 1-5, DOI: 10.1109/ITHET.2018.8424780.

[16] Secretaría de la OMPI, "Documento de Referencia sobre Patentes y Nuevas Tecnologías", Organización Mundial de la Propiedad Intelectual – OMPI, SCP/305, pp. 5-7, Junio 2019. [En línea]. Disponible en: https://www.wipo.int/edocs/mdocs/scp/es/scp_30/scp_30_5.pdf

[17] J. Brownlee (2020, Agosto 20). A Gentle Introduction to the Rectified Linear Unit (ReLU). [En línea]. Disponible en: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/

[18] J. Brownlee (2020, Setiembre 15). Your First Deep Learning Project in Python with Keras Step-By-Step. [En línea]. Disponible en: https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/

[19] S. Ricart, Rubén A., S. Gil-Guirado, M. Hernández-Hernández, Antonio M., J. Olcina-Cantos (2020). "Could MOOC-Takers' Behavior Discuss the Meaning of Success-Dropout Rate? Players, Auditors, and Spectators in a Geographical Analysis Course about Natural Risks", pp. 7, 1-18.

[20] Python. [En línea]. Disponible en: https://www.python.org/

[21] Google Colab. [En línea]. Disponible en: https://colab.research.google.com/

[22] Tensor Flow. [En línea]. Disponible en: https://www.tensorflow.org/

[23] Scikit-learn. [En línea]. Disponible en: https://scikit-learn.org/

[24] Microsoft Word. [En línea]. Disponible en: https://www.microsoft.com/es-ww/microsoft-365/word?rtc=1

[25] D. Calvo. "Aprendizaje Supervisado", Marzo 2019. [En línea]. Disponible en: https://www.diegocalvo.es/aprendizaje-supervisado/

[26] A. Moujahid, I. Inza, P. Larrañaga, "Tema 5. Clasificadores K-NN", Departamento de Ciencias de la Computación e Inteligencia Artificial Universidad del País Vasco–Euskal Herriko Unibertsitatea, pp. 1-8. [En línea]. Disponible en: http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t9knn.pdf

[27] J. Orellana, "Arboles de decisión - Parte I", Arboles de decision y Random Forest, Noviembre 2018, [En línea]. Disponible en: https://bookdown.org/content/2031/arboles-de-decision-parte-i.html

[28] J. Martínez, "¿Qué es un Random Forest?", Setiembre de 2009, [En línea]. Disponible en: https://www.iartificial.net/random-forest-bosque-aleatorio/#:~:text=Un%20Random%20Forest%20es%20un,distintas%20porciones%20de%2

0los%20datos.&text=Esto%20hace%20que%20cada%20%C3%A1rbol,datos%20para%20un%20mismo%20problema.

[29] C. Isidro, “Propuesta de un método basado en Deep Learning para Learning Analytics en MOOCs”, Trabajo de Fin de Máster, Universidad Autónoma de Madrid, Madrid, 2017.

[30] tyagikartik4282 (2019, Julio 07). SVM Hyperparameter Tuning using GridSearchCV | ML. [En línea]. Disponible en: https://www.geeksforgeeks.org/svm-hyperparameter-tuning-using-gridsearchcv-ml/